**3GPP TSG-SA5 Meeting #155 *S5-243212***

**Jeju, Korea (Republic Of), 27th May 2024 - 31st May 2024**

**Source: Ericsson**

**Title: Rel-19 pCR TR 28.871 Schema retrieval enhancements**

**Document for: Approval**

**Agenda Item: 6.19.8**

# 1 Decision/action requested

***Approve the proposal.***

# 2 References

[1] 3GPP TR 28.871 Study on Service Based Management Architecture enhancement phase 3

[2] 3GPP TS 28.532 Generic management services

[3] IETF RFC6022 YANG Module for NETCONF Monitoring

[4] IETF RFC8525 YANG Library

# 3 Rationale

A key enabler to for data-driven producers, and consumers, is the ability to retrieve the supported schemas from the network.

As of Rel-18 there is limited supported defined to retrieve such information from the network in a standardized way. Only the ‘root’ schema retrieval is currently defined, per [2], Annex

# 4 Detailed proposal

**First change**

# 5.a Schema retrieval enhancements

5.a.1 Description

This enhancement proposes methods to support retrieval of schema files/modules.

5.a.2 Potential requirements

**REQ-Schema-retrieval -1:**  The MnS Producer shall support advertising its supported schema files/modules.

**REQ-Schema-retrieval -2:**  The schema advertisement shall support files/modules located at any location specified by the MnS Producer.

**REQ-Schema-retrieval -3:**  The MnS Producer shall support retrieval of all files/modules.

**REQ-Schema-retrieval -4:**  The MnS Producer shall support retrieval of specific schema files/modules.

5.a.3 Potential solutions

This enhancement proposes methods to support retrieval of schema files/modules.

#### 5.a.3.1 Potential solution#1, Schema list at known location

This enhancement proposes methods to support advertisement and retrieval of schema files/modules reusing some portions of RFC6022 [3] provides as a common stage 2 in NRM:

1. Add NRM to define each available schema. Define *Schema* IOC with properties:
	1. Schema id (identifier, e.g. file/module name)
	2. Schema version (based on versioning information in the file/module)
	3. Schema format (Enum: yang, yaml)
	4. Schema namespace (URI)
	5. Schema location (file URI)
	6. Supported feature list (if applicable)
	7. Indicator of “implemented” vs. “imported” (if applicable)
2. Advertisement of supported schema:
	1. Define a specific location from which the list of supported schema is available. Similar to the ../schemas/.. tree defined in [3], clause 2.1.3 this would be a predefined location from which any consumer can query the list.
	2. Each schema would be an entry in a ‘list’ of schema available at this location
3. Schema retrieval:
	1. Consumer uses the information in the schema list entries to retrieve the schema file(s) of interest.
	2. The actual method to retrieve them (e.g. HTTP, sftp) is defined by the solution and indicated in the <protocol> portion of the schema location URI.

#### 5.a.3.1 Potential solution#2 Schema lists as capabilities

This enhancement is similar to #1, but instead of placing the schema at a known location the schema files would be added to the capabilities advertised by specific MnS and/or managed entities. I.e. the schema file details would be data in the capabilities, not individual MOIs per schema file.

#### 5.a.3.2 Potential solution#3 Solution set specific solutions

This enhancement proposes different methods based on solution set. No stage2 or common NRM is required.

1. OpenAPI: predefine a resource end point from which the schema list can be queried. Define as a tree allowing single/multiple entries to be queried.
2. NETCONF: implement per RFC6022 or RFC8525

5.a.4 Evaluation of potential solutions

TBD

**2nd change**

# 5.b Schema reference enhancements

5.b.1 Description

The current ProvMnS.yaml contains all possible schema entries when only a subset may actually be supported by a particular management system. Additionally, a management system may require adjustment of the list of supported schema at runtime, e.g. when new NFs with different schema and/or versions are introduced.

In both cases above the current hard-coded ProvMnS which contains all possible entries despite their applicability, and which cannot be adjusted at runtime is not sufficient.

5.b.2 Potential requirements

**REQ-Schema-retrieval -1:**  The MnS Producer shall support flexible extension of the required schema files.

5.b.3 Potential solutions

This enhancement proposes replacing the current predefined schema references in ProvMns.yaml with an extensible mechanism allowing the ProvMnS to dynamically adjust to the required schema list at runtime.

#### 5.b.3.1 Potential solution#1, Update ProvMnS OpenAPI definition to be more generic

Currently the ProvMnS.yaml includes the following specific list of schema entries:

    Resource:

      oneOf:

        - type: object

          properties:

            id:

              type: string

            objectClass:

              type: string

            objectInstance:

              $ref: 'TS28623\_ComDefs.yaml#/components/schemas/Dn'

            attributes:

              type: object

          additionalProperties:

            type: array

            items:

              type: object

          required:

            - id

        - anyOf:

            - $ref: 'TS28623\_GenericNrm.yaml#/components/schemas/resources-genericNrm'

            - $ref: 'TS28541\_NrNrm.yaml#/components/schemas/resources-nrNrm'

            - $ref: 'TS28541\_5GcNrm.yaml#/components/schemas/resources-5gcNrm'

            - $ref: 'TS28541\_SliceNrm.yaml#/components/schemas/resources-sliceNrm'

            - $ref: 'TS28536\_CoslaNrm.yaml#/components/schemas/resources-coslaNrm'

            - $ref: 'TS28312\_IntentNrm.yaml#/components/schemas/resources-intentNrm'

            - $ref: 'TS28104\_MdaNrm.yaml#/components/schemas/resources-mdaNrm'

            - $ref: 'TS28105\_AiMlNrm.yaml#/components/schemas/resources AiMlNrm'

            - $ref: 'TS28538\_EdgeNrm.yaml#/components/schemas/resources-edgeNrm'

            - $ref: 'TS28317\_RanScNrm.yaml#/components/schemas/resources-RanScNrm'

Proposal is to update this to remove the references, and allow the ProvMnS to provide its list of supported schema (i.e. all supported schema as advertised/available for the MnS):

Resource:   # Resource must be compliant with application/vnd.3gpp.object-tree-flat+json or application/vnd.3gpp.object-tree-hierarchical+json as per NRM definitions (28.541), See 32.158 $4.3.1, $ 6.1.4 for more info

         type: object

          properties:

            id:

              type: string

            objectClass:

              type: string

            objectInstance:

              $ref: 'TS28623\_ComDefs.yaml#/components/schemas/Dn'

            attributes:

              type: object

          additionalProperties:

            type: array

            items:

              type: object

5.b.4 Evaluation of potential solutions

Solution #1 is not recommended for the reason it introduces a missing connection between ProvMnS schema and supported feature schema. A solution which addresses this concern, especially one considering Forge requirements, is preferred.